Week 2. Big Data Analytics - `data.frame` manipulation

Hyeonsu B. Kang
hyk149@eng.ucsd.edu

March 2016

The following material was recreated from `http://goo.gl/fT7KI` and `http://goo.gl/lbICv`.

1 Data Frame

A data frame is used for storing data tables. It is a list of vectors of equal length. For example, the following variable `df` is a data frame containing three vectors `n`, `s`, `b`.

```
n = c(2, 3, 5)
s = c("aa", "bb", "cc")
b = c(TRUE, FALSE, TRUE)
df = data.frame(n, s, b)
```

1.1 Indexing

There are built-in data frames in R that we can use. Here, let's use one called `mtcars`.

```
mtcars
```

The top line of the table, called the **header**, contains the column names. Each horizontal line afterward denotes a **data row**, which begins with the name of the row, and then followed by the actual data. Each data member of a row is called a **cell**. To retrieve data in a cell, we would enter its row and column coordinates in the single square bracket `[]` operator. The two coordinates are **separated by a comma**. In other words, the coordinates begins with row position, then followed by a comma, and ends with the column position. The order is important.

Here is the cell value from the first row, second column of `mtcars`.

```
mtcars[1, 2]
```

Moreover, we can use the row and column names instead of the numeric coordinates.

```
mtcars["Mazda RX4", "cyl"]
```

Subsetting an entire column of the data frame is also possible. For example, we can only take the `cyl` column of the data frame by

```
mtcars[, "cyl"]
```

or by using the index of that column

```
mtcars[, 2]
```

or equivalently, we can use the double bracket notation:

```
mtcars[["cyl"]] # note the double square brackets
```

or we can use a dollar sign notation as well:

```
mtcars$cyl
```

Subsetting multiple columns at once is also possible. How would you retrieve both `cyl` and `hp` columns? You can do similar operations to the rows as well. Retrieving the first 5 rows from `mtcars` can be done as follows:

```
mtcars[1:5]
```

To know the number of columns or rows of the data frame, use `ncol()` and `nrow()` functions:

```
ncol(mtcars)
nrow(mtcars)
```

Similarly we can retrieve rows from a data frame. To access row 24 of `mtcars`

```
mtcars[24,]
```

or equivalently,

```
mtcars["Camaro Z28",]
```

What happens with the following code?

```
mtcars["Camaro Z28"]
```

To retrieve more than one rows, use a numeric index vector that contains the corresponding row indices:

```
mtcars[c(3, 24),]
```

1.2 Logical Indexing

We can retrieve rows with a logical index vector. In the following vector `L`, the member value is `TRUE` if the car has automatic transmission, and `FALSE` if otherwise.

```
L = mtcars$am == 0
L
```

What would happen if we used `L` as a row index vector?

```
mtcars[L,]
```

Among these observations, how can we retrieve the gas mileage data?

```
mtcars[L,]$mpg
```

Among the observations, how can we retrieve both the `wt` value is bigger than `3.0` and the `cyl` value is `6`?

```
M = mtcars$wt > 3.0
N = mtcars$cyl == 6
O = M & N # vectorized operation
mtcars[L,]$mpg
```

Note the vectorized operation `&` in `O = M & N`.

1.3 Practice

Problem 1. Can you retrieve observations from `mtcars` where the `disp` value is greater than or equal to `150.0` and less than `400.0`, the value of `gear` is 4, and the value of `mpg` is bigger than `17.5`?
Problem 2. Can you retrieve observations from `mtcars` where the `wt` value is bigger than `1.5` and smaller than `3.75`, or the `carb` value is `1, 2, 3, or 6`?
Problem 3. Can you retrieve observations that satisfy both of the conditions mentioned above?
Problem 4. Can you retrieve observations that do **not** satisfy the first condition or that do **not** satisfy the second condition? How many observations are there?